UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LEVEL II (12)

AD A099255

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AIM 629

2. GOVT ACCESSION NO.: AD-A099255

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): (6) Active Touch Sensing,

5. TYPE OF REPORT & PERIOD COVERED: (9) Memorandum rept.

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Daniel (10) William D. Hillis

8. CONTRACT OR GRANT NUMBER(s): (15) N00014-80-C-0505

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (11)

11. CONTROLLING OFFICE NAME AND ADDRESS:
Advanced Research Projects Agency
1400 Wilson Blvd
Arlington, Virginia 22209

12. REPORT DATE: Apr[redacted]81

13. NUMBER OF PAGES: 37

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
Office of Naval Research
Information Systems
Arlington, Virginia 22217 (12) 39

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):

Distribution of this document is unlimited.

(14) AI-M-629

DTIC ELECTE MAY 22 1981 E

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):

Touch
Tactile Sensor
Tactile
Tendon
Robot
Finger

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

The mechanical hand of the future will roll a screw between its fingers and sense, by touch, which end is which. This paper describes a step toward such a manipulator--a robot finger that is used to recognize small objects by touch. The device incorporates a novel imaging tactile sensor--an artificial skin with hundreds of pressure sensors in a space the size of a finger tip. The sensor is mounted on a tendon-actuated mechanical finger, similar in size and range of motion to a human index finger. A program controls the finger,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED 407482

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

Best
Available
Copy

using it to press and probe the object placed in front of it. Based on how the object feels, the program guesses its shape and orientation and then uses the finger to test and refine the hypothesis. The device is programmed to recognize commonly used fastening devices-nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo 629 April 1981

ACTIVE TOUCH SENSING

WILLIAM DANIEL HILLIS

The mechanical hand of the future will roll a screw between its fingers and sense, by touch, which end is which. This paper describes a step toward such a manipulator-- a robot finger that is used to recognize small objects by touch. The device incorporates a novel imaging tactile sensor- an artificial skin with hundreds of pressure sensors in a space the size of a finger tip. The sensor is mounted on a tendon-actuated mechanical finger, similar in size and range of motion to a human index finger. A program controls the finger, using it to press and probe the object placed in front of it. Based on how the object feels, the program guesses its shape and orientation and then uses the finger to test and refine the hypothesis. The device is programmed to recognize commonly used fastening devices- nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws.

This report describes research done at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. The research was supported in part by IBM, under an agreement with the Massachusetts Institute of Technology Laboratory for Computer Science. Support for the Artificial Intelligence Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agnecy of the Department of Defense under contract with the Office of Naval Research contract N00014-80-C-0505. The author is supported by a fellowship provided by the Fannie and John Hertz Foundation

81 5 15 181

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo 629 April 1981

ACTIVE TOUCH SENSING

WILLIAM DANIEL HILLIS

The mechanical hand of the future will roll a screw between its fingers and sense, by touch, which end is which. This paper describes a step toward such a manipulator-- a robot finger that is used to recognize small objects by touch. The device incorporates a novel imaging tactile sensor- an artificial skin with hundreds of pressure sensers in a space the size of a finger tip. The sensor is mounted on a tendon-actuated mechanical finger, similar in size and range of motion to a human index finger. A program controls the finger, using it to press and probe the object placed in front of it. Based on how the object feels, the program guesses its shape and orientation and then uses the finger to test and refine the hypothesis. The device is programmed to recognize commonly used fastening devices- nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws.

This report describes research done at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. The research was supported in part by IBM, under an agreement with the Massachusetts Institute of Technology Laboratory for Computer Science. Support for the Artificial Intelligence Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agnecy of the Department of Defense under contract with the Office of Naval Research contract N00014-80-C-0505. The author is supported by a fellowship provided by the Fannie and John Hertz Foundation

81 5 15 181

CONTENTS

1. Introduction **3**

2. The Sensor **4**

2.1 Method of Construction 4
2.2 Scanning the Array 8
2.3 Performance 11

3. The Mechanical Finger **14**

3.1 Relating Motor Motion to Joint Motion 17
3.2 Dynamics 19
3.3 The Human Finger 19

4. The Program **23**

4.1 The Domain: Nuts and Bolts 25
4.2 Representation: Describing How Something Feels 26
4.3 Implementation 27
4.4 Limitations, What Needs to be Done Next? 32

5. Acknowledgments **35**

1. Introduction

A dexterous robot manipulator must be able to feel what it is doing. The mechanical hand of the future will roll a screw between its fingers and sense, by touch, which end is which. This paper describes a step toward such a manipulator-- a robot finger that is used to recognize small objects by touch. The device incorporates a novel imaging tactile sensor- an artificial skin with hundreds of pressure sensors in a space the size of a finger tip. The sensor is mounted on a tendon-actuated mechanical finger, similar in size and range of motion to a human index finger. A program controls the finger, using it to press and probe the object placed in front of it. Based on how the object feels, the program guesses its shape and orientation and then uses the finger to test and refine the hypothesis. The device is programmed to recognize commonly used fastening devices- nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws.

The paper is divided into three main sections. The first is a description of the tactile sensor array-- how it is constructed and what it can do. The second section describes the mechanical finger and how it is controlled. The final section of the main portion of the paper is a description of a program that uses the finger and sensor to recognize small objects. Finally, there is an appendix, describing some related work on tendon hands and arms.

Accession For
A

2. The Sensor

The touch sensor is a monolithic array of 256 tactile sensors that fits (appropriately) on the tip of a finger. This is comparable to the resolution of the human forefinger. Each sensor has an area of less than one hundredth of a square centimeter and gives an independent analog indication of the force over its surface in the range of 1 to 100 grams. The array is scanned one column at a time to minimize the number of connecting wires. The sensor is rugged, flexible, and has a skin-like texture.

2.1 Method of Construction

The touch array has two conductive components: a flexible printed circuit board and a sheet of anisotropically conductive silicone rubber (ACS). The ACS has the peculiar property of being electrically conductive along only one axis in the plane of the sheet. The printed circuit board is etched into fine parallel lines, so it too conducts in only one dimension. The two components are placed into contact with the lines on the printed circuit board perpendicular to the ACS axis of conduction. The contact points at each intersection of the perpendicular conductors form the pressure sensors.

The device must also include a separator to pull the conducting layers apart when pressure is released. The sensitivity and range of the sensor depend largely on the construction of this intervening layer. For a large pressure range the best separator I tested was the woven mesh of a nylon stocking. For high sensitivity, a separator may be deposited directly onto ACS by spraying it with a fine mist of non-conductive paint. The conductive rubber presses through the separator so that the area of contact, and hence the contact resistance, varies with the applied pressure.

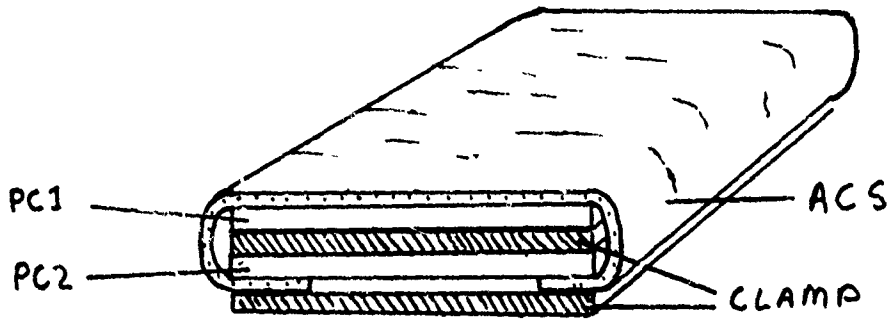
The pressure/resistance relationship is non-linear, as shown in Figure 5. I do not have a model of the contact mechanism that quantitatively explains the change in resistance with applied pressure, however Figure 2. illustrates a plausible qualitative model. Pressure on the elastomeric

ACS deforms the material around the separator, allowing it to contact the metal below. Larger pressures results in more deformation and larger contact areas. If the resistance of the contact is proportional to the contact area, the contact resistance will be inversely related to the applied pressure. For the object recognition application, the non-linear response of the sensor was not a significant drawback.

The ACS itself is constructed of layers of silicone rubber impregnated with either graphite or silver, alternating with similar non-conductive layers. Each layer is approximately 250 microns thick. The layers are oriented at right angles to the plane of the sheet. In its normal commercial application, ACS is sliced into strips and used to make contact between printed circuit boards. The linear resistivity, in the conducting direction, is on the order of kilohms per centimeter for graphite-impregnated ACS. This is inconveniently high for building large sensors, and I was able to lower it to approximately 100 ohms per centimeter by electroplating it with gold. It is possible to plate only over the conductive silicone, so that the cross resistance remains essentially infinite. Silver-impregnated silicone rubber has a substantially lower bulk resistance, but I was unable to obtain the material in the proper form. The minimum resolution of commercially available ACS is about 50 lines per centimeter.

Wires are connected to the edges of the printed circuit board by soldering. The ACS is mounted in such a way that its edges fold around the printed circuit, where they are pressed against contact fingers on the other side (see Figure 1). A compound sensor with high range and good sensitivity may be constructed by placing a high range (nylon mesh) sensor behind a sensitive one. In this case, I eliminated the flexible circuit board in the front layer, so that the center layer of ACS was shared between the two arrays. (I have constructed a sensor of this type, but it was not used in conjunction with the mechanical finger.)

Fig. 1. Mechanical Drawing of Touch Sensor



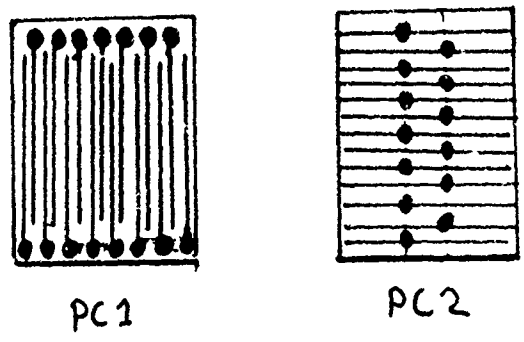
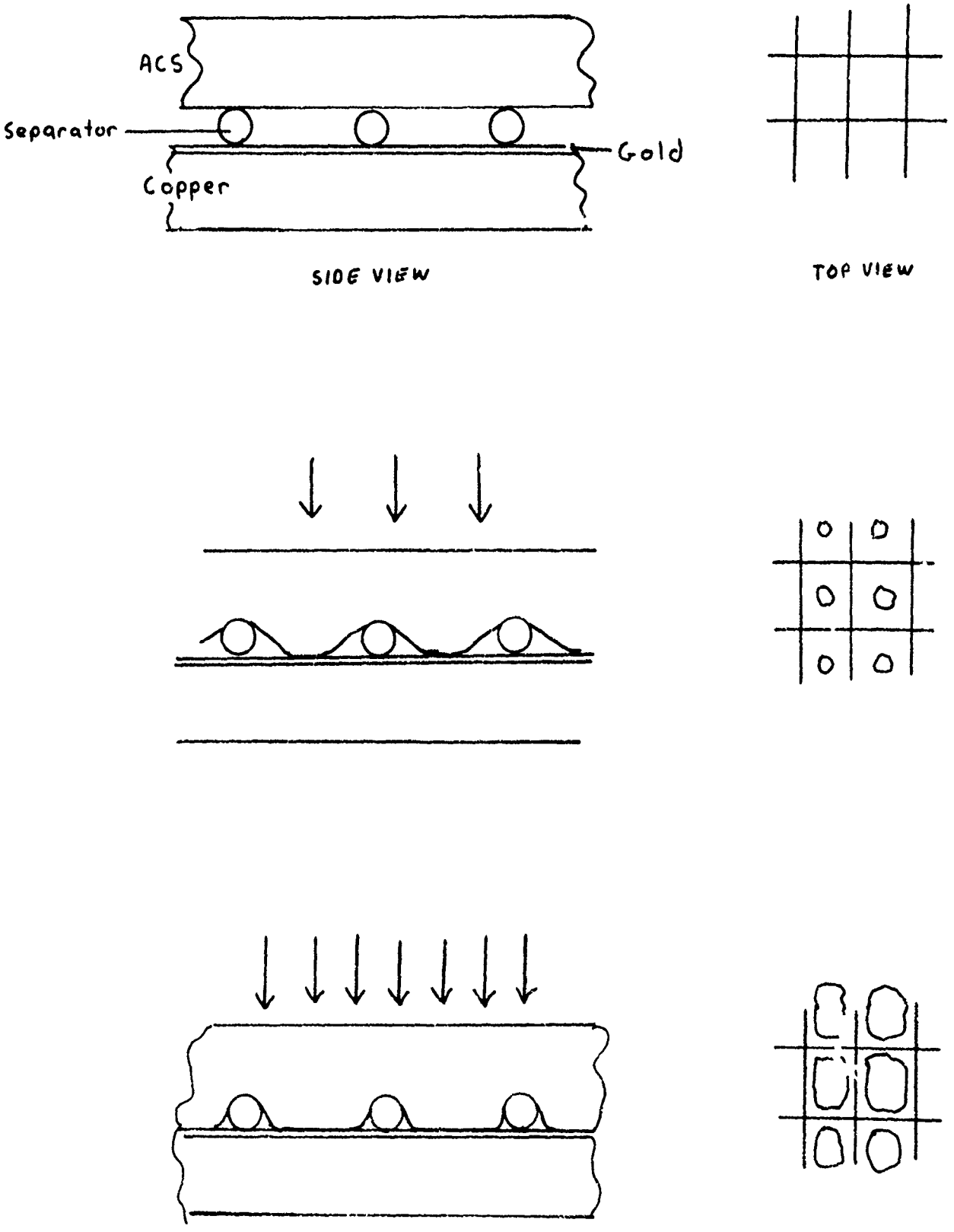


Fig. 2. Contact Restistance Changes with Changing Area

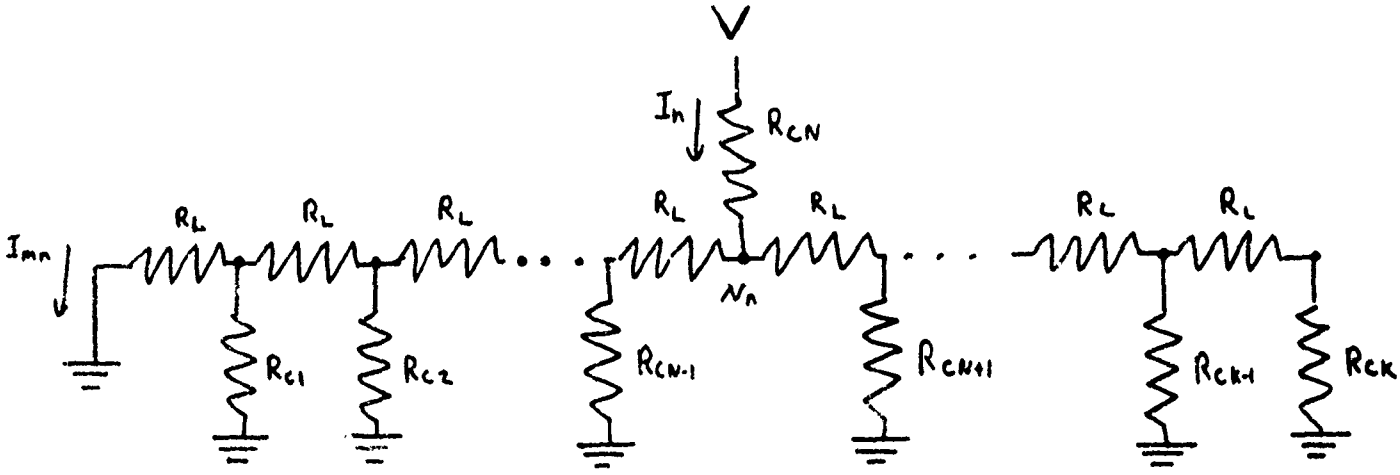


2.2 Scanning the Array

Attaching wires only at the edges of the array reduces the number of necessary connections. This is important given the limited space of a mechanical finger. (My 256 cell sensor used 32 wires, size #42, stranded. The resulting cable is less than 3 mm. in diameter.) The array is scanned by applying a voltage to one column at a time and measuring the current flowing in each row. A potential problem with this method is the introduction of phantom tactile images. When multiple points are activated simultaneously, it may appear that untouched point are also conducting. This is the analog version of the crosspoint problem in xy-scanned keyboards: if three out of four switches on a rectangle are closed, the fourth appears to be also. This happens because the path through the other three connections is electrically in parallel with the ghost connection. In keyboards, it is usually avoided by putting a diode at each point of intersection. This could be done for touch array also, but it would add considerably to the complexity of the device and it might also introduce undesirable mechanical stiffness. With resistive contacts it is theoretically possible to compute the actual resistances from the measured resistances by solving N equations in N unknowns [4], but the technique tends to amplify errors due to inaccuracy of measurement, noise, and resistance along the conductive axis. Other researchers [9,1,8] have avoided the problem by attaching a separate wire to each sense point. This is impractical for high resolution arrays and, again, it limits mechanical flexibility.

Instead I used the scheme illustrated in Figure 4. It is similar to the voltage mirror approach suggested by Purbrick [7]. A fixed voltage is placed on the column of interest, while all other columns are held at ground potential to ground out any alternate paths. The rows are all held to ground potential also, by injecting whatever current is necessary to cancel the current injected by the active column. The value of the resistance of a crosspoint is inversely proportional to the current that is necessary to pull the corresponding row to ground potential. By this method, extraneous columns are at the same potential as the rows (ground), so no current will flow through the unmeasured crosspoints. The holding currents depend only on the column drive voltages and

Fig. 3. Electical Model of One Row



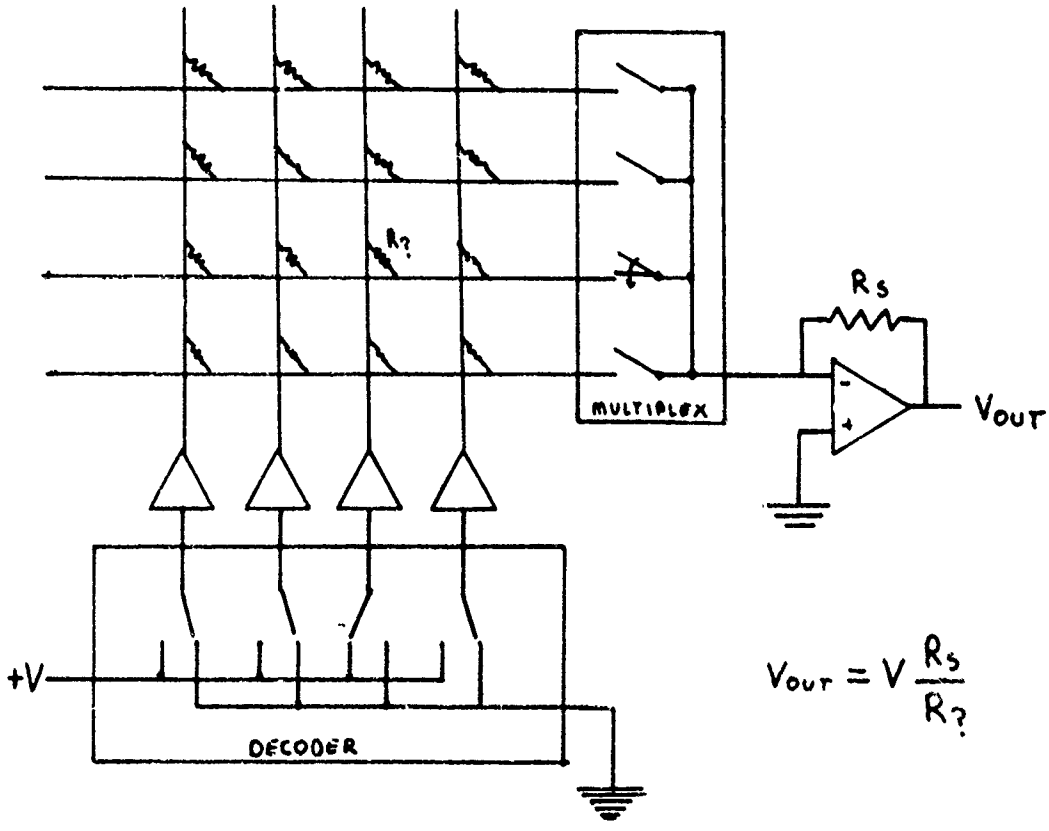
the resistances in question. The entire array is scanned by measuring one column at a time, as described above.

The method described is valid only if the crosspoint resistances are high compared to the linear resistances of the row and column lines, otherwise it is not possible to hold an entire row or column at a fixed potential. This effect may be understood by referring to the electrical model of a single row illustrated in Figure 3. The applied voltage (V) and the linear resistance R_L are known, but the unknown resistance (R_{Cn}) can not be determined unless the potential at node N_n is known. This potential may be computed, in time proportional to the number of nodes, by first measuring the unknown resistances near the edge. The resistance of all the the unknown contacts may be determined by computing the successive two-port parameters of the subnetworks toward the edge of the unknown node.

$$Z_0 = 0$$

$$G_0 = 1$$

Fig. 4. Scanning the Array



$$R_n = \frac{V - I_{mn} G_{n-1}(Z_{n-1} + R_L)}{I_n}$$

$$Z_n = \frac{R_n(Z_{n-1} + R_L)}{Z_{n-1} + R_n + R_L}$$

Where V is the applied voltage, I_n and I_{mn} the measured currents, Z_n the input impedance and G_n is the voltage transfer ratio of the network to the left of R_n.

For large arrays it may actually be necessary to compute the resistances as shown, but as long a the linear resistance is low, as compared with the contact resistance, this is not necessary. If the measured values are used directly, then the worst case error for a row of N elements is-

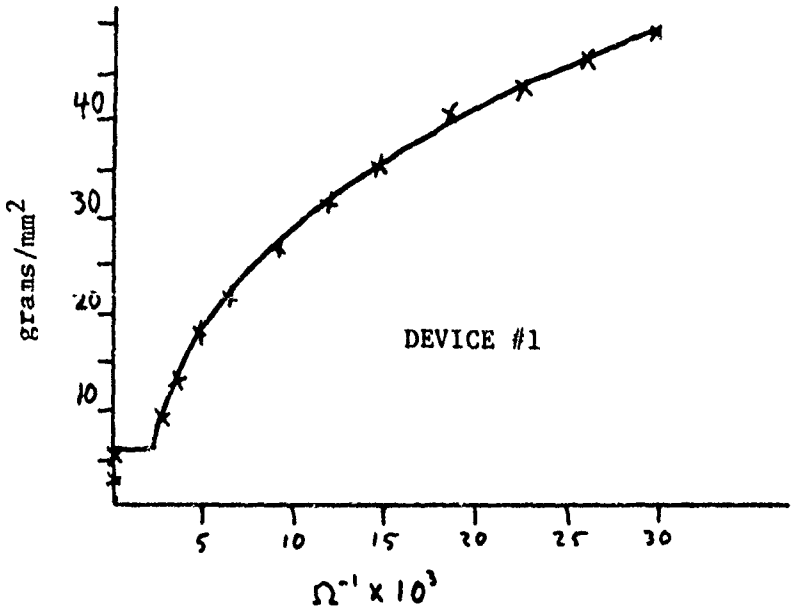
$$\frac{R_{measured}}{R_{actual}} = \frac{R_n + NR_L}{R_n}$$

This is easy to determine because the worst case occurs when all contacts, except for the one being measured, are open. Other contact closures will only lower the potential of node N_n, increasing the accuracy of the measurement. The error may also be reduced by a factor of two by making contact at both ends of the row.

2.3 Performance

I constructed several sensory arrays with varying range and resolution. Figure 5. shows pressure/resistance curves for two representative devices. Device #1 has a sprayed separator (approximately 10^4 dots per square centimeter). The separator of device #2 is nylon mesh (Leggs, Extra Sheer). The ACS used in device #1 was plated with gold on the contact side. All devices showed good mechanical durability and, after an initial settling period, stable electrical characteristics. (The first prototype, almost a year old, shows no noticeable change in contact resistance.) The highest resolution device (#1) was a 16 by 16 array, one centimeter in area. This is the sensor used with the finger. Sample images of the top of a screw, an electronic connector, a 1/8 inch ring, and a cotter pin are shown in Figure 6.

Fig. 5. Performace Curves of Two Sensors



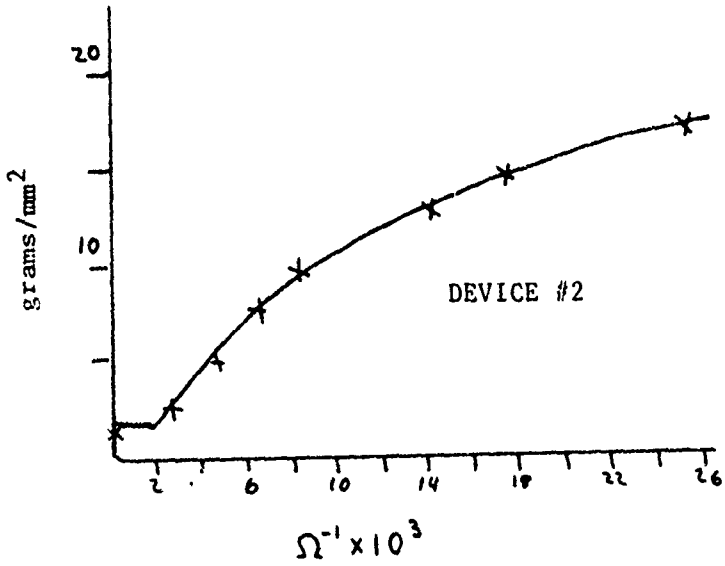
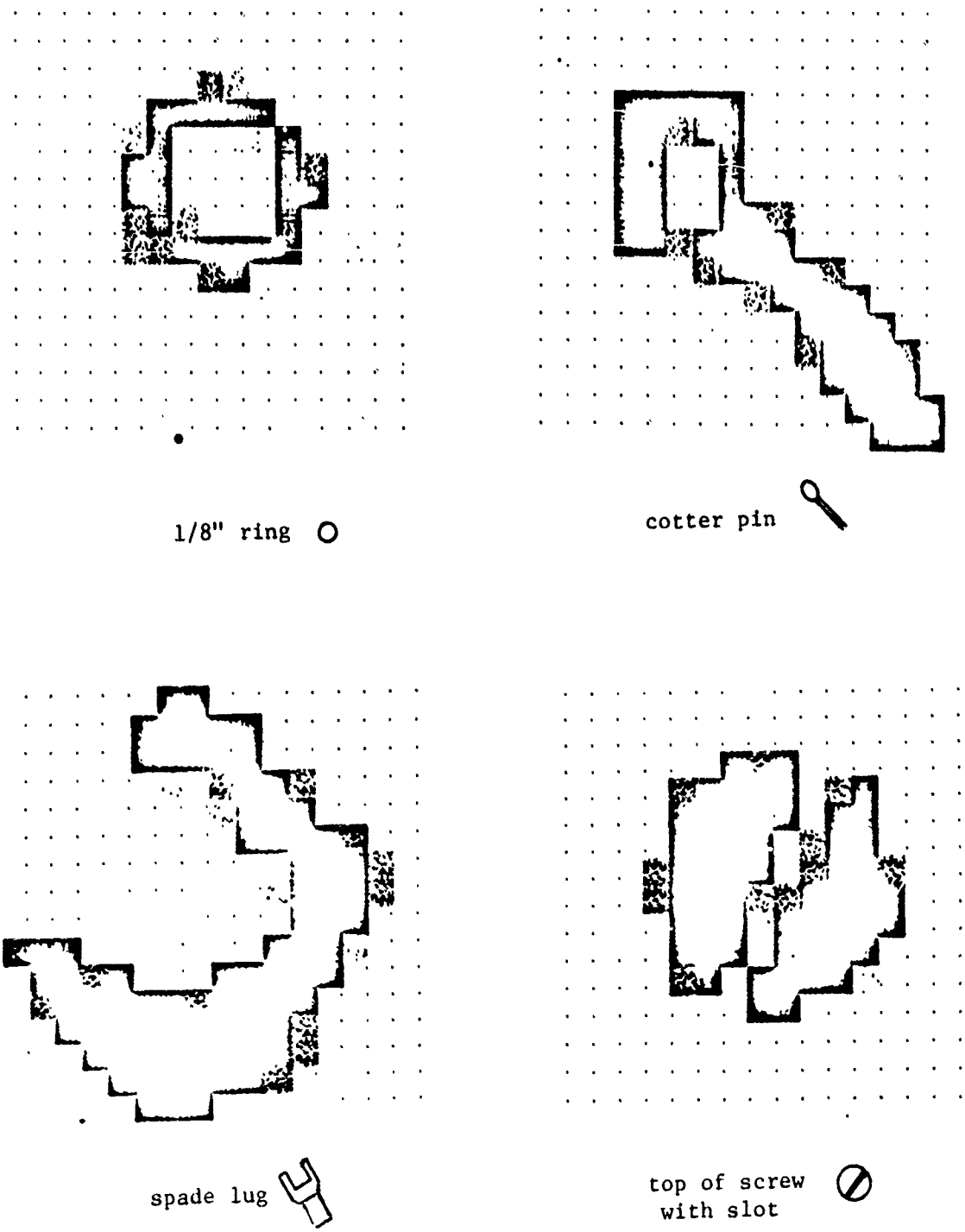


Fig. 6. Sample Tactile Images from the Sensor
(actual size objects are shown below images)



3. The Mechanical Finger

The finger, on which the touch sensor is mounted, has approximately the same shape, size and range of motion as the human forefinger. Like the human finger, it has three joints: a two degree of freedom pivot at the base and two single degree of freedom hinge joints. These joints are controlled by four pairs of tendons, which are driven from four electric motors mounted behind the base of the finger. Torque and position are measured only at the motors, so that joint torques and angles must be computed as described below. The finger and associated motors are mounted on a fixed base, with a small platform extending just below the finger, on which objects may be placed for testing.

The tendons of the finger are arranged in opposing pairs-- one bends the joint, the other straightens it. A system of pulleys keeps the total length of each pair constant. This allows both ends of a tendon pair to be driven from a single motor. The lever arm of the tendon pulling against the joint is kept constant over all angles by winding the tendon over a pulley fixed to the joint.

The body of the finger is constructed of flat aluminum sections which are held together by press-fitted dowel pins. All non-fixed pulleys are mounted on precision ball bearings. This is not really mechanically necessary, but the reduced friction makes the computation of joint torques more accurate.

The tendon material is braided Kevlar, 1/64th inch diameter, coated with polyurethane for abrasion resistance. Kevlar was chosen over steel because the minimum bending radius is smaller for a given diameter of cable. It is the minimum bending radius of the tendon material that limits the minimum size of the finger. If a more flexible tendon material was used, the design could easily be scaled to one half its current size, even using commercially available ball bearings. Smaller fingers would require special bearings, for example, the jewel bearings that are commonly used in watches.

The total length of each tendon pair must be kept constant so that the pair may be

driven by a single motor. This is accomplished by winding the tendons around pulleys in opposite directions as they pass through a joint. When the joint bends, one tendon winds around the pulley as much as the other unwinds. This guarantees that the sum of the lengths of the two tendons remains constant. Each tendon has a stiff spring inserted along its length to set the common-mode tension. The lengths of the two terminal tendon pairs are not actually invariant over changes in the horizontal direction of the proximal joint. This slight length change is compensated for by stretching in the tendon springs. The position coupling would be insignificant except that it complicates the computation of forces on the finger. A change in the total length of the tendons causes the spring to stretch, and this in turn puts additional forces on the joint. The effect could be eliminated completely by adding another pulley stage, or by using constant force springs, instead of conventional springs, to set the common mode tension on the pair. A constant force spring does not change the apply force as it is stretched, so a length change would make no difference.

The motors used to drive the tendon were ring-shaped moving-coil torque motors. This type of motor was originally developed for gyroscope applications where precise control of torque is critical. The motors have a large number of poles and commutator points to reduce cogging effects. The particular motors used to drive the finger have a torque constant of 2.6 ounce inches per amp, plus or minus 2%. The internal winding resistance of these motors is about 10 ohms. An important property of these motors is their relatively large torque constant. This allows the tendons to be driven from a pulley directly attached to the shaft of the motor, without intervening gearing. Any such gearing would have introduced problems with back-lash, moment multiplication, etc. The primary disadvantage of the motors was their relatively high turning mass. This problem might have been reduced by the use of printed circuit motor, but at the cost of additional turning friction.

The angle of the motors is sensed by a coaxial potentiometer. The pulley sizes of the motor are such that the full range of joint motion may be achieved with less than a full rotation

any of the motors. This allows the use of high-resolution single-turn potentiometers. The resistive element in these potentiometers is made of conductive plastic, rather than wound wire, for increased resolution. The motors have no external velocity sensors. Instead, velocity is computed from the motor's induced voltage, as described later.

3.1 Relating Motor Motion to Joint Motion

The tendons in the mechanical finger are arranged to give independent control of the torque and angle of each joint. Optimally the finger would be controlled by four motors, with the angle and torque of each joint corresponding directly to the angle and torque of one of the motors. In practice, this is difficult to realize because the tendon controlling one joint must pass through other joints on the way to its insertion. The mechanical designs which are easiest to implement have arbitrary non-linear couplings between the motors and the joints. I chose a compromise: Each joint has associated with it a *primary motor*. The angle of the joint is a linear function of the primary motor angle and the angles of the primary motors of the intervening joints. (It is actually possible, at least in principle, to design a tendon finger with no joint interactions. What is required is a means of passing a tendon through a bending joint without changing its length. Just going through the joint's axis of rotation will almost work, except that all tendons have a minimum bending radius. A cable inside an incompressible sheath, with no pulley, is another possibility. The most successful robot hands so far [5,6] have been of this second type.)

The linearity constraint makes computation of the joint angles a matter of simple matrix multiplication:

$$m = Mj$$

Where m is the motor angle vector, j is the joint angle vector, and M is a lower triangular matrix. Assuming that the joints are numbered starting from the base of the finger, the (i,j)th entry of the matrix is the ratio of the radii of pulley for the ith tendon in the jth joint and the ith

motor pulley.[1]

Because the ith tendon must pass through all joints less than i and none greater than i, the matrix must be lower triangular, with no zeros in the bottom. Since all the diagonal elements are non-zero, M is invertible. This is important because the inverse matrix is used by the controller to compute the motor positions for a given set of joint positions. For the pulleys actually used the matrix was:

$$\begin{pmatrix} 1.00 & 0 & 0 & 0 \\ 0.67 & 0.67 & 0 & 0 \\ 0.32 & 0.48 & 0.48 & 0 \\ 0.32 & 0.32 & 0.32 & 0.48 \end{pmatrix}$$

A similar equation relates the joint and motor torques. In this case the terms of the matrix can depend on the angle of the joints, so there is a different matrix for each finger position. In this case the matrix is upper triangular: changing the torque in the primary motor of a joint will affect the torques in all proximal joints. This is the opposite of what happens with angles; A change in a joint's angle affects the joints which come after it, a change in a joint's torque affects the joints which come before it.

The diagonal terms of the torque matrix are the reciprocals of the diagonal terms in the position matrix, that is, they are the ratios of the radii of the motor pulleys to the joint pulleys. The off-diagonal terms depend on the angles of the joints and the mechanical layout of the device. The finger was designed to keep these terms positive at all angles. This has control advantages. Consider the typical control problem of holding a joint angle constant while a varying force is applied at the tip. To stabilize the finger, an increase in force in the terminal joint must be accompanied by an increase in force in the proximal joints. This is exactly the effect of having positive off-diagonal terms in the upper half of the torque matrix. The magnitude of these terms is such that they provide useful mechanical feedforward.

---

1 Actually, the effective radii of the pulleys must be used for this calculation- The radius of the pulley plus the radius of the tendon

3.2 Dynamics

In analyzing the dynamics of the mechanical finger it is realistic to consider only the mass of the motors. The mass of the finger itself is small by comparison. The motors may be controlled in one of two modes: position mode or torque mode. For the purpose of analyzing the motor driving circuitry, the motor may be modeled electrically as an ideal servo motor in series with a resistor. This means that the torque on the motor is proportional to the current passing through it, and that the angular velocity of the motor is proportional to the voltage across the terminals minus a correction factor (proportional to the current) for the series resistance of the motor. In this application the current is controlled and the voltage is measured to determine the angular velocity.

Equivalent drive circuits are shown in Figure 8. In torque control mode, the current through the motor is sensed as a voltage across a resistor, and servoed to a level proportional to the desired torque. In position control mode, the velocity of the motor is proportional to the positional error. This is accomplished by supplying a current proportional to the difference between voltage across the motor and the positional error voltage. The voltage is proportional to velocity, and the current proportional to torque, which is proportional to angular acceleration. Therefore, the acceleration is proportional to the velocity error. For simplicity, the series resistance correction circuitry is not shown.

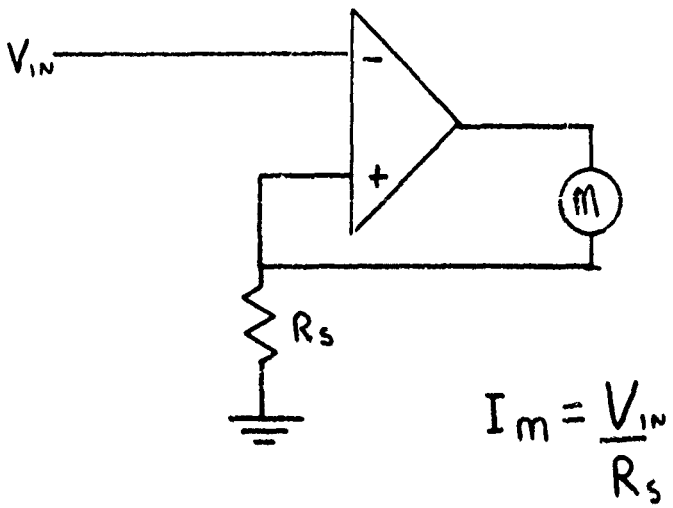
3.3 The Human Finger

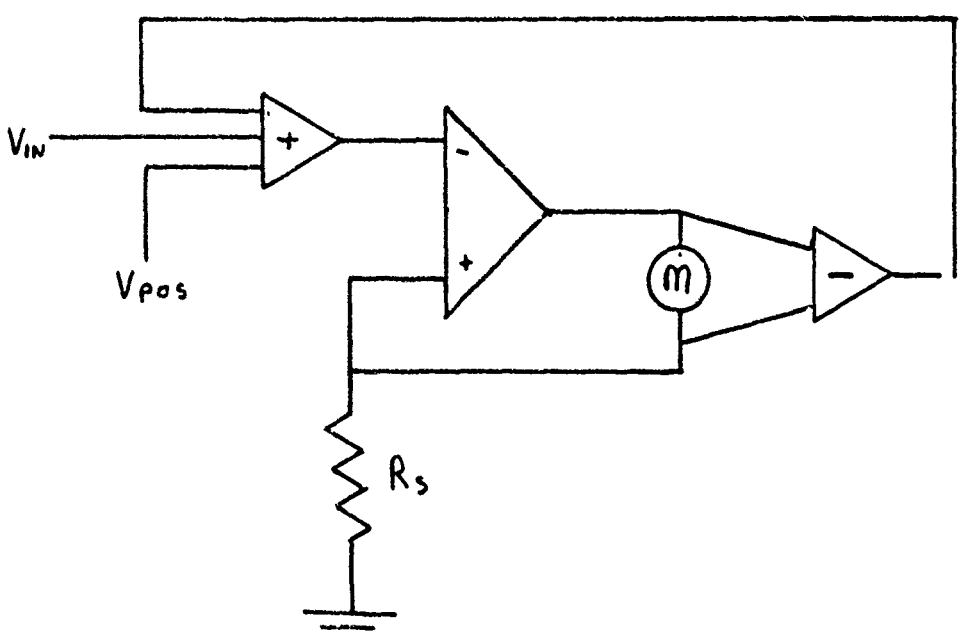
For comparison, it is useful to consider the source of inspiration for this design-- the human finger. [1] There are three bones in each finger. The joint at the base of the finger is a two degree of freedom ball and socket joint, with the socket on the finger bone. Twisting is

---

1 I am pleased to say that I had an opportunity to personally verify these well know anatomical facts by dissecting a human cadaver I would like to thank Tufts Medical School for lending me a hand

Fig. 8. Circuit Models for Motor Drivers

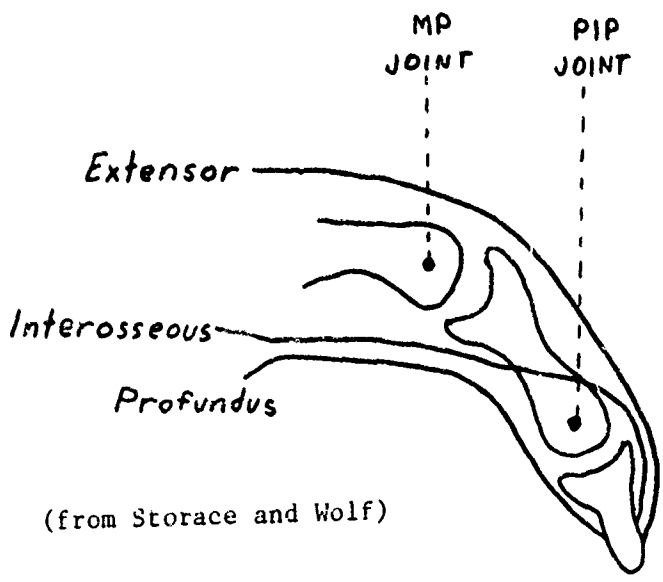




constrained by the ellipsoid shape of the joint; the ball is wider than it is tall. Lateral motion is about 15 degrees in either direction, and is due partly to the motion of the metacarpal bones in the hand. The finger can flex almost 90 degrees toward the palm, and extend about 10 degrees away from it. The second and third joints each have only a single degree of freedom, constrained by a grooved ball and mating projection in the socket. The range of motion is about 120 degrees in the medial joint and 90 degrees in the terminal joint. Both joints may be hyperextended by a few degrees. The terminal bone is much smaller than the others and tapers to a point.

Many of the muscles that control the fingers are located in the forearm. They are connected to the finger bones by long flat tendons running inside sheaths. There are also small muscles in the palm that originate at the metacarpals and insert directly into the phalanges. Exactly how these actuators work together to to move the finger is not well understood; it has been the topic of at least two recent doctoral theses [10,3]. The problems are that the actuators are not used in simple opposing pairs, the way they attach to the bones is difficult t nodel, and the

Fig. 9. Model of the Human Finger



(from Storace and Wolf)

joints themselves are complicated. Figure 9 shows a simplified mechanical model of the joints and actuators. For a more accurate mechanical model of the human finger see [12] or [11].

4. The Program

The tendon finger and the touch sensor are controlled by a tactile recognition program. The program uses the finger to press and probe the object placed in front of it. Based on how the object feels, the program guesses the shape and orientation of the object. The device is programmed to recognize commonly used fastening devices- nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws. The program is written in Lisp and runs on a specially augmented Lisp Machine [13], with independent micro-processors to control low-level input/output functions.

4.0.1 Why Touch is Easier than Vison

In writing this simple tactile recognition system I have retraced, in both technique and spirit, the steps of early researchers in machine vision. There is good reason to believe that these simple techniques have a better chance of working in the tactile domain than they did in the visual. For one thing, there are far fewer data to be analyzed than in a visual image. This means that even with a high resolution tactile array, complex processing may be performed in real time. Another factor is that collection is more readily controlled. Since placement and pressure of the fingertip are controlled by the program, analyzing a tactile image is like analyzing a visual image with controlled background, illumination, and point of view.

There is also a third factor responsible for making tactile recognition the easier task: The properties that we actually measure are very close, in kind, to the properties that we wish to infer. In vision, it is only possible discover mechanical properties (shape, orientation, absolute position) by deducing them from optical properties (shading, projection, reflectivity). In touch, we measure mechanical properties directly.

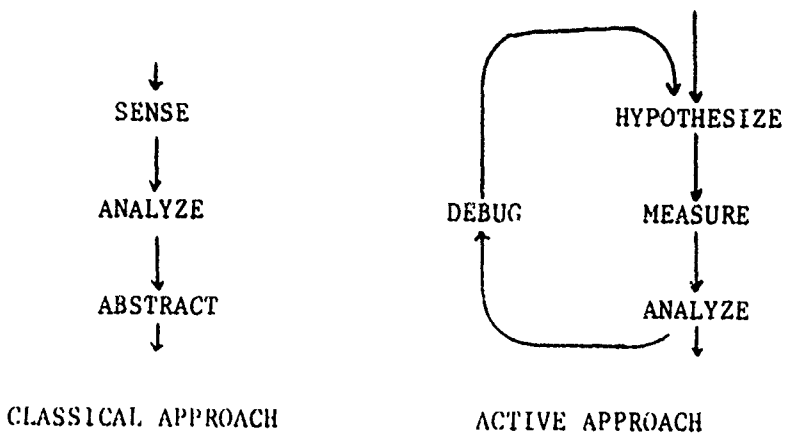
4.0.2 Active Sensing

I would like to contrast here two possible approaches to sensory recognition, in touch or in vision. The first is what I will call the classical approach. In the classical approach we start an image of some sort, extract features of the image and then, from the features, abstract some kind of a model of what is shown in the image. The classic approach is bottom-up or data driven. The second approach, which I shall call the active approach, is top-down or knowledge driven. When taking the active approach, we begin with a theory of what is in the image. Based on that theory we make measurements or perform experiments to test validity of the hypothesis. The results of the theory are then analyzed using the same analytic techniques as the classical approach, and based on the result of the analysis, the hypothesis may be modified or confirmed. If it is modified, we try experiments to test the new hypothesis, and so on, until we arrive at a conclusion that agrees with the data. The two approaches are represented schematically in Figure 8.

The choice of top-down vs. bottom-up was a source of extended controversy among early researchers in machine vision. The general consensus today seems to be that processing at

---

Fig. 10. Two Approaches to Recognition



the lowest levels is bottom-up and processing at higher levels in top-down. This is consistent with what I am calling the active approach.

In the tactile domain, I believe that the active approach will yield the best results. For one thing, the information in a single image is often insufficient for recognition of the object. Moving the finger to make different measurements is the best way of collecting enough data, and in order to decide how to move the finger, the program must have some expectation of what object it is feeling. The program should operate at all times with a hypothesis of what it is feeling. It recognizes the object by actively probing it with the finger, modifying its internal "hallucination" of the object to conform with the measured reality. This is what I mean by *active* touch sensing.

4.1 The Domain: Nuts and Bolts

For the purpose of testing the sensor, the finger and the recognition techniques, I choose a restricted range of test objects which the program would be expected to recognize. Specifically, I choose a set of commonly used mechanical fastening devices such as screws and dowel pins. One advantage of these objects is that they are important for potential industrial applications of robotics. Recognition of fasteners and determination of their position and orientation when they are grasped by a manipulator is an important industrial problem. It is also a problem that is unlikely to be solved by machine vision because the hand obscures the object and because forces cannot be seen. In this domain tactile and visual sensing would complement each other well: vision for locating objects and measuring their absolute position, touch for sensing local shape, orientation, and forces once they are grasped.

The particular objects chosen for study were machine screws, set screws, flat washers, lock washers, dowel pins and cotter pins. They were chosen because they have simple shapes that are easy to represent and easy to distinguish. Restricting the range of possible objects to such a small and easily distinguishable subset makes the recognition task less difficult, but it also

introduces the possibility that he recognition methods used are not really generally applicable. This may be the case here, but I have been aware of the danger and have tried to implement the various algorithms is such a way that they may be extended to a larger range of objects.

The actual objects used for testing were all small (not more than 1/2 inch in any dimension). In general, the smallest commonly used size in a given category was used for testing. For example, the machine screw was size #0-80 by 3/8 inch and the cotter pin was 1/2 inch long by 1/16 inch diameter. Using such small objects allowed the entire image o be read in one impression of the sensor. This avoided the problem of coordinating multiple sensor impressions into a single tactile image. In addition, it made for more impressive demonstrations.

4.2 Representation: Describing How Something Feels

In designing the recognition program, I began by developing a simple language for describing the tactile properties of an object. The source of ideas for this first-pass tactile description language was introspection. What properties do I notice when I feel an object?. After a few minutes of rolling the test objects under my (biological) finger, I noticed three categories of features that I was measuring. These are parameters I choose for representing the feel of an object. Here is a list of the of the parameters-

SHAPE- The general shape of the object. For the fastener micro-world there are only two possible shapes: ROUND and LONG. The shape may therefore be determined directly from the aspect ratio of the image.

BUMPS- The locations of local pressure anomalies. These locations are expressed in object relative coordinates, for example, in coordinates relative to the major and minor axes of the image. Bump may include both positive bumps that stick out, and negative bump (holes) that stick in. (In the implemented program, only the sign and position of a bump are considered significant for matching purposes. There is no intensity information.)

STABILITY- This property indicates how easy it is to roll the object in various directions. (in the program, two boolean values represent the stability of an object. These indicate whether or not the object rolls freely along each of the primary axes.)

These three parameters proved sufficient to distinguish between any of the objects in the test set. Figure 9 shows which properties each of the objects exhibits.

4.3 Implementation

Given a representation of the objects in question, how do you write a program that measures the relevant features? This section describes a first pass that I have made at writing such a program.

The routines described below have all been written and tested, but the complete sequence of sub-steps has not yet been strung together into a single program. I will begin with a description of

Fig. 11. Table of Objects

		Shape	Bumps	Rolls
	Machine Screw	LONG	+	YES
	Dowel Pin	LONG	0	YES
	Cotter Pin	LONG	-	NO
	Set Screw	ROUND	0	YES
	Lock Washer	ROUND	+	NO
	Flat Washer	ROUND	0	NO

the hardware on which the program runs, because some of it was specifically designed for this application.

4.3.1 The Hardware

The control tasks are divided between a Lisp Machine and five Z80 micro-processors. Most of the code is written in Lisp, and runs on the Lisp Machine. The micro-processors are each dedicated to a simple task with real time constraints. One scans the tactile image array, while others take specifications for joint motions, and execute them in real time. The memories of the micro-processors are directly accessible by the Lisp Machine. This shared memory provides a convenient interface between the two levels of processing. The output of the tactile array, for example, is available as a Lisp array object. A lisp program may treat this array just as any other array: the memory sharing is invisible to the software.

Another example of the use of shared arrays is in the control of joint motions. A lisp program computes the desired joint motions, or chooses them from a library of precomputed motions. Each motion is stored as a vector of joint angles or torques, representing the successive states of the joint over a period of time. These vectors are one dimensional arrays, shared between the two machines. The actual control of the motors is handled by the micro-processor, using the specifications in the shared arrays. Real time control would be difficult to realize on the Lisp Machine alone, because of virtual memory swapping, and multiple-process scheduling.

4.3.2 The Top Level Recognition Loop

Written in LISP, the top level of the recognition program looks like this-

```
(defun recognize ()                                  ;debug loop
  (do ((setq assumed-image (best-match)))            ;hypothesize
      ((test-assumptions))))                         ;test

(defun test-assumptions ()
  (do ((l (funcall real-image ':unknown-attributes) (cdr l)))
      ((null l) t)
    (or (similarp (funcall real-image (car l)) (funcall assumed-image (car l)))
        (return nil)))) ;try to verify each assumption
```

This program is just a translation of the "Active Sensing" flowchart shown is Figure 10. It begins by assuming that the object is whatever best matches the information that it has so far. This is the "hypothesize" step. It then tests assumptions that it has made about the object, and unless they are all correct, repeats the process from the beginning. Testing the assumption may involve moving the finger to roll or probe the object, or may just involve making a computation from already collected data.

Flow of control in this process depends on what information is needed. When a property is queried, if it is not already known, it is computed or measured. In the process of computing the value, the program may make queries about properties, which in turn may need to be computed. For example, if we ask an object's shape, and it is not known, then it must be computed from the object's dimensions along the primary axes. If these axes are not known, they must be computed from the image of the object. If no image has been read in, the finger must be pressed against the object. And so on. This is "call by need" control flow. It prevents information from being measured or computed unless it is actually needed in the recognition process.

4.3.3 Crunching an Image

After an image is read in, it must be processed to determine the object's location, the primary axes, and the location of any bumps. The first step in processing the tactile image is the elimination of unwanted detail. This is accomplished by convolving the image with a simple pulse function. The image is contrast-enhanced to two bits per pixel by comparing it to fixed threshold values. If the offset pressure was chosen properly (see below) the four possible pressure values correspond to background, depressions, primary figure and bumps.

```
(defun recognize ()                                  ;debug loop
  (do ((setq assumed-image (best-match)))            ;hypothesize
      ((test-assumptions))))                         ;test

(defun test-assumptions ()
  (do ((l (funcall real-image ':unknown-attributes) (cdr l)))
      ((null l) t)
    (or (similarp (funcall real-image (car l)) (funcall assumed-image (car l)))
        (return nil)))) ;try to verify each assumption
```

This program is just a translation of the "Active Sensing" flowchart shown is Figure 10. It begins by assuming that the object is whatever best matches the information that it has so far. This is the "hypothesize" step. It then tests assumptions that it has made about the object, and unless they are all correct, repeats the process from the beginning. Testing the assumption may involve moving the finger to roll or probe the object, or may just involve making a computation from already collected data.

Flow of control in this process depends on what information is needed. When a property is queried, if it is not already known, it is computed or measured. In the process of computing the value, the program may make queries about properties, which in turn may need to be computed. For example, if we ask an object's shape, and it is not known, then it must be computed from the object's dimensions along the primary axes. If these axes are not known, they must be computed from the image of the object. If no image has been read in, the finger must be pressed against the object. And so on. This is "call by need" control flow. It prevents information from being measured or computed unless it is actually needed in the recognition process.

4.3.3 Crunching an Image

After an image is read in, it must be processed to determine the object's location, the primary axes, and the location of any bumps. The first step in processing the tactile image is the elimination of unwanted detail. This is accomplished by convolving the image with a simple pulse function. The image is contrast-enhanced to two bits per pixel by comparing it to fixed threshold values. If the offset pressure was chosen properly (see below) the four possible pressure values correspond to background, depressions, primary figure and bumps.

Next the aspect ratio and major and minor axes of object are determined. Computation of the aspect ratio begins by first locating the center of the activated area. For this purpose all non-zero pixels are taken to be part of the object. The center and the point farthest away from it determine the major axis of the object. The minor axis is taken to be perpendicular to this. These axes provide an an object-relative coordinate system in which it is possible to specify, roughly, the location of bumps and depressions in the image. The bounding rectangle of the object is taken to be the smallest rectangle, with edges parallel to the axes, which contains the image. The aspect ratio of the object is taken to be the aspect ratio of its bounding rectangle.

4.3.4 Moving the Finger

If the image read in is not satisfactory, it is possible to move the finger and read another-- we do not have to rely on first impressions. An important part of the image analysis involves moving the finger so that an optimal image is sensed. The offset pressure, for example, may be adjusted in this manner. Optimally, most of the touched area activated the mid-range of the sensor, allowing bumps and depressions to be easily detected. This is accomplished by reading in an image, computing the median pressure of all points above the noise threshold, and readjusting the finger pressure appropriately. This may be repeated several times until an acceptable offset pressure is achieved.

The finger is also moved to measure the stability (resistance to roll) of an object. To measure the object's stability in a given direction, the object is pressed between the finger and the supporting surface by applying a fixed force on the object normal to the plane of the surface. The finger is then moved laterally in the desired direction. (The supporting surface should have a high coefficient of friction on the object to prevent sliding.) The stability of the object is indicated by the amount of force necessary to move the finger.

4.3.5 The Matcher

During the hypothesize step, the program must determine which of the objects it knows best matches the known data. For a small possibility set, such as the fasteners, it is not really important that this is done well. For a large set of possible objects the quality of the matcher may be a determining factor in the speed of recognition. When hypothesis is chosen by selecting the possibility that best matches the information given, usually the choice that has the largest number of features in common with the known facts is the best choice. In a system with a large number of parameters other factors may also be taken into consideration. For one thing some features may be more important than others, either in general or for that particular possibility. Also, the features themselves may not exactly match-- a bump may be too large, a shape distorted. In cases such as this we wish to give the possibility only partial credit for a feature match.

The most obvious way to implement such a matcher would be a numerical scoring system with the weighting factors for feature importances and partial matches. I avoided such a solution for two reasons. First, there would have to be a degree of arbitrariness in assigning the numbers: Is a circle a 50% match to a hexagon? Is shape 2.5 times as important as texture, or only twice? It is unwise to trust the sums and products of numbers if the numbers themselves are chosen arbitrarily. The second objection is more of a philosophical one- converting a complex set of symbolic structures into a single number throws away too much information, too quickly. Of course, this information must eventually be lost- the matcher must terminate by selecting a single item. But the pruning can be, and is, controlled in a more reasoned manner.

The implemented matcher takes two possibilities at a time, and compares them on a feature by feature basis. If, for a particular feature, both items match the image to about the same degree, the information is ignored. If one of the items is clearly a better match, the feature is counted in favor of the appropriate item. This procedure is repeated for each feature and then the features themselves are compared in a similar manner. A feature counted toward one item will cancel with a feature counted toward another, if they are of approximate importance.

I am assuming that in a large AI system the best match would be computed in parallel. Parallel marker propagation schemes, such as the one proposed by Fahlman [2], would do such a task well. One important assumption, even for the parallel case, is that the binary comparison operator is transitive. Without this constraint it would be necessary to compare each possible pair of items, a task which grows as the square of the number of items.

The transitivity of the predicate described above can be easily demonstrated, given the transitivity of individual feature comparisons. Assume there exist three items A,B, and C such that A > B and B > C. Let f(x,y) be the set of features counted in favor of x when compared with y. Since the individual feature comparisons are transitive, f(A,C) = (f(A,B) ∪ f(B,C)) and f(C,A) = (f(C,B) ∪ f(B,A)). If ≻ is the feature set comparison predicate (the second stage of the algorithm above), then A > B implies f(A,B) ≻ f(B,A. Also, for any sets a,b,c and d such that a ≻ b and c ≻ d, it must be that (a ∪ c) ≻ (b ∪ d), because features that cancel in the individual sets will also cancel in the union. The assumptions , A > B and B > C, imply f(A,B) ≻ f(B,A) and f(B,C) ≻ f(C,B), and by the union rule (f(A,B) ∪ f(B,C)) > (f(C,B) ∪ f(B,A)). This may be rewritten as f(A,C) ≻ f(C,A), which is the criterion for A > C. Therefore, the matching predicate is transitive.

This matcher is really overkill for a possibility set of six objects with three parameters each, but it may be necessary if the program is to be extended to a large range of objects.

4.4 Limitations, What Needs to be Done Next?

One should not be too impressed by a program that distinguishes between six objects on the basis of three parameters. If only a single bit of information was derived from each parameter, it should be enough to recognize at least eight objects. In the future, tactile recognition programs will have more complex and more precise representations of tactile images. In this last section, I would like to mention three improvements that I believe are just around the corner.

The first is texture recognition. The resolution of the tactile array sensor, while high, is grossly insufficient for measuring textural differences between, say, paper and glass. Texture

sensing, require the detection of bulk effect of multiple surface features. It is most easily accomplished sliding something over the surface and noticing the pattern of vibrations, in much the same way that we slide a phonograph needle over a record. In fact, I have done some preliminary experiments using just that, a phonograph needle. Sensor of the future may use embedded piezo-electric devices, or it may be possible to use the ACS directly as sort of a carbon microphone. However the information is derived, it must be processed into a useful characterization of the texture of the surface. One of the things we are interested in is the intensity and periodicity of the signal. These features may be seen directly in the frequency domain. Texture processing may bear more similarity to the analysis of sounds than to the analysis of visual images.

Another difference between paper and glass is that glass feels cold. This is not actually because that glass is lower in temperature, but because it is a better conductor of heat and so it is able to more quickly carry away the heat generated by the body. I have constructed a small thermal conductivity sensor that works on this principal. In the sensor, a resistive heating element is sandwiched between two temperature sensitive current sources. Any difference in the temperature of the two sensors is indicated by an easy to measure difference in the currents. The sensor is designed to be mounted on the finger in such a way that one temperature sensor may contact the device being tested. As the heat is drawn from the object into the object, a difference in temperatures will develop. The primary disadvantage of this first prototype is that it is large (0.1 inches x 0.3 inches x 0.2 inches) resulting in a relatively high thermal mass. This limits both the response time and the minimum size of object which may be usefully tested. I believe the time is ripe for more work in this area.

The third area which shows immediate potential for further research is the coordination of multiple tactile images into a global picture. I deliberately avoided this problem in my studies by choosing a small object that could be read in a single impression. Restricting the range possible of objects to this degree imposes limitations that may be unacceptable outside of the laboratory

environment. I have done no work in this area, but I believe it to be an approachable, solvable problem.

I am enthusiastic about the future prospects of automated tactile sensing. The field is begging for more work. What has been described here-- the sensor, the finger, the program-- only scratches the surface of what is possible. The mechanical hand of the future will have a sense of touch.

5. Acknowledgments

I would like to thank have contributed work, ideas and enthusiasm. They are Mike Brady, Max Behenski, Tom Callahan, James Davis, Fred Drenckhahn, Gary Drescher, Richard Greenblatt, Greg Gargarian, John Purbrick, Jerry Roylance, Gerald Sussman, John Hollerbach, Michael Dertouzos, Ed Hardebeck, Henry Minsky, Margaret Minsky, Lisa Schostakovich, Laurel Simmons, Latanya Sweeney, Patrick Winston and most importantly, Marvin Minsky.

1. Broit, M. The Utilization of an "Artificial Skin" sensor for the Identification of Solid Objects. Proc. 9th Int. Symp. on Industrial Robotics, March 1979

2. Fahlman, Scott, NETL: A System for Representing and Using Real-World Knowledge. MIT Press 1979

3. Fischer, G.W. A Treatise on the Topographical Anatomy of the Long Finger and a Biomechanical Investigation of its Interjoint Movement, Ph.D. Thesis, University of Iowa, Iowa City, 1969

4. Larcombe, M.H.E., Tactile Sensors Sonar and Parallax Sensors for Robot Applications. Third Conference on Industrial Robot Technology, C3, March 1976

5. Okada,Tokuji. Object Handling System for Manual Industry. IEEE Transactions on Systems, Man, and Cybernetics. Vol SMC-9, No. 2, February 1979

6. Okada,Tokuji and Tsuchiya,Seiji. On a Versatile finger System, Proc. 7th Int. Symp. Industrial Robots, October 1977

7. Purbrick, John A., A Force Transducer Employing Conductive Silicone Rubber. First Robot Vision and Sensors Conference

8. Harmon, L. D.,Touch Sensing Technology.

9. Stojiljkovic, Z. and J. Clot, Integrated Behavior of Artificial Skin. IEEE Transaction on Biomedical Engineering.., Vol BME-24, No. 4 July 1977, 396-399

10. Storace, Anthony, A treatise on the biomechanics if the human finger. Polytechnic Institute of New York New York, 1977

11. Storace, Anthony and Wolf, Barry, Functional Analysis of the Role of Finger Tendons, J. Biomechanics Vol. 12, 1979

12. Thomas, Donald, Long, Charles and Landsmeer, J.M.F. Biomedical Considerations of Lumbricalis Behavior in the Human Finger, J. Biomechanics, Vol. 1, 1968, 107-115

13. Weinreb, Daniel and Moon, David. Lisp Machine Manual, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1979